



Aula 17 - Modelo Publisher - Subscriber - MQTT

Módulo de Internet das Coisas

Prof^a. Nídia Glória da Silva Campos



UNIVERSIDADE
ESTADUAL DO CEARÁ



MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO





IA

Objetivos da Aula

- Apresentar o protocolo MQTT
- Demo-Lab: MQTT no Packet Tracer
- Atv 11 - Dados de um sistema monitoramento de presença para armazenamento em nuvem



IA

Cenários IoT trazem novos desafios [1]

- Requer um modelo em tempo-real, orientado a eventos
- publicação de informação um-para-muitos
- escutar os eventos à medida que acontecem
- enviar pacotes pequenos de dados para dispositivos pequenos
- envio de dados seguro sobre redes inseguras



Para dispositivos móveis e IoT...

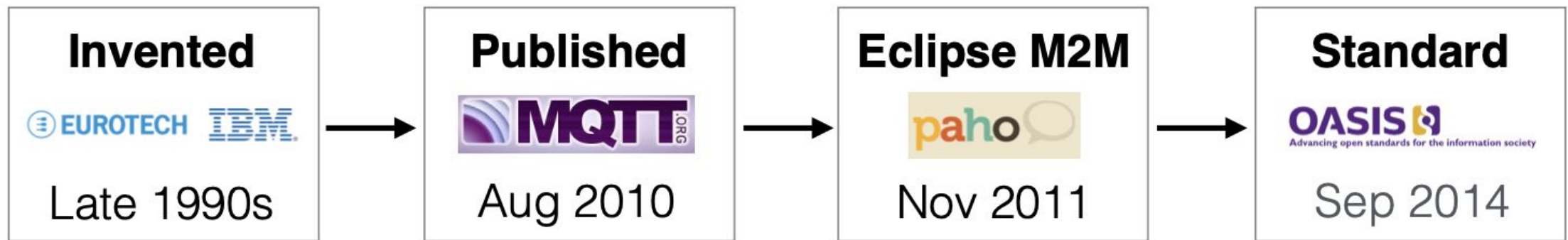
mensagens são muitas vezes > HTTP request/response



IA

MQTT: um protocolo leve para mensageria IoT [1]

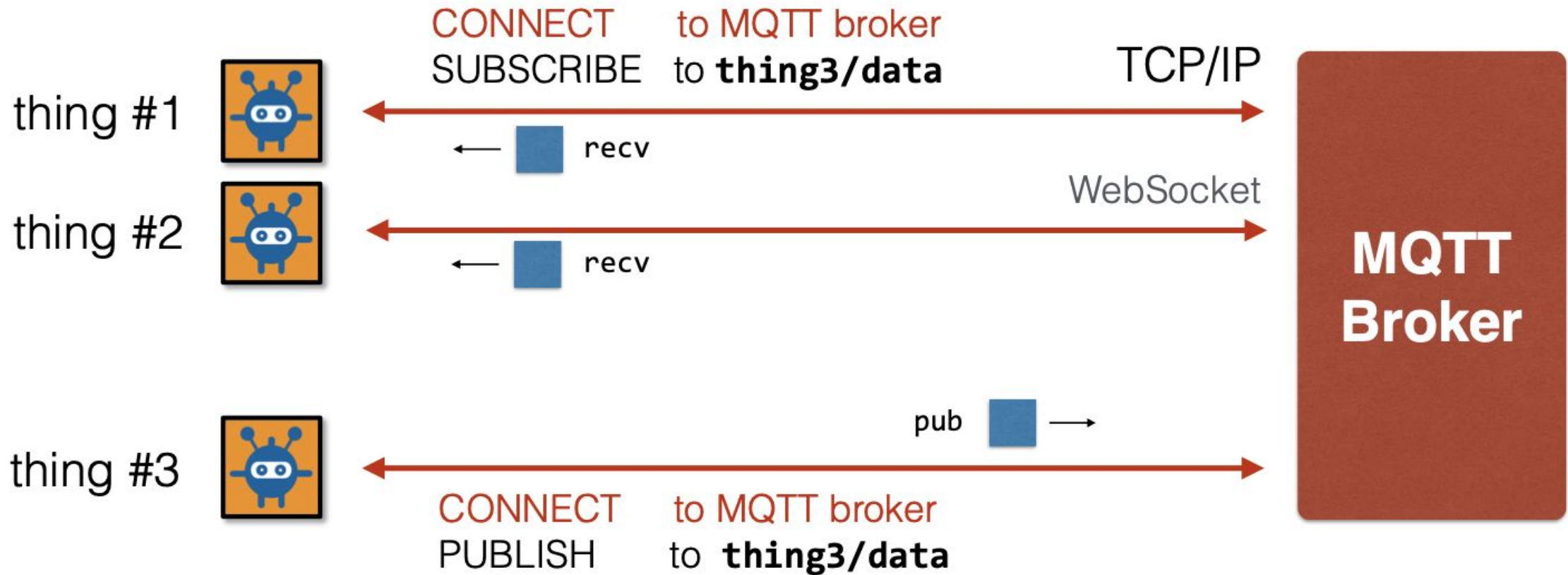
- aberto
 - especificação aberta, padrão
- leve
 - overhead mínimo, formato eficiente, clientes minúsculos (kb)
- confiável
 - QoS para confiabilidade em redes não-confiáveis
- simples
 - especificação de 43 páginas, connect+publish+subscribe





IA

MQTT: comunicação bidirecional, assíncrona [1]





IA

MQTT: publish-subscribe [2]

- publisher: cliente que transmite uma mensagem de dados
- subscriber: cliente que recebe mensagens de dados
- tópico: consiste nos dados transmitidos
- broker: servidor que conecta os clientes e filtra dados
 - por assunto: um tópico e/ou branches de tópicos
 - por conteúdo: dados transmitidos
 - por tipo: filtros criados pelos próprios clientes

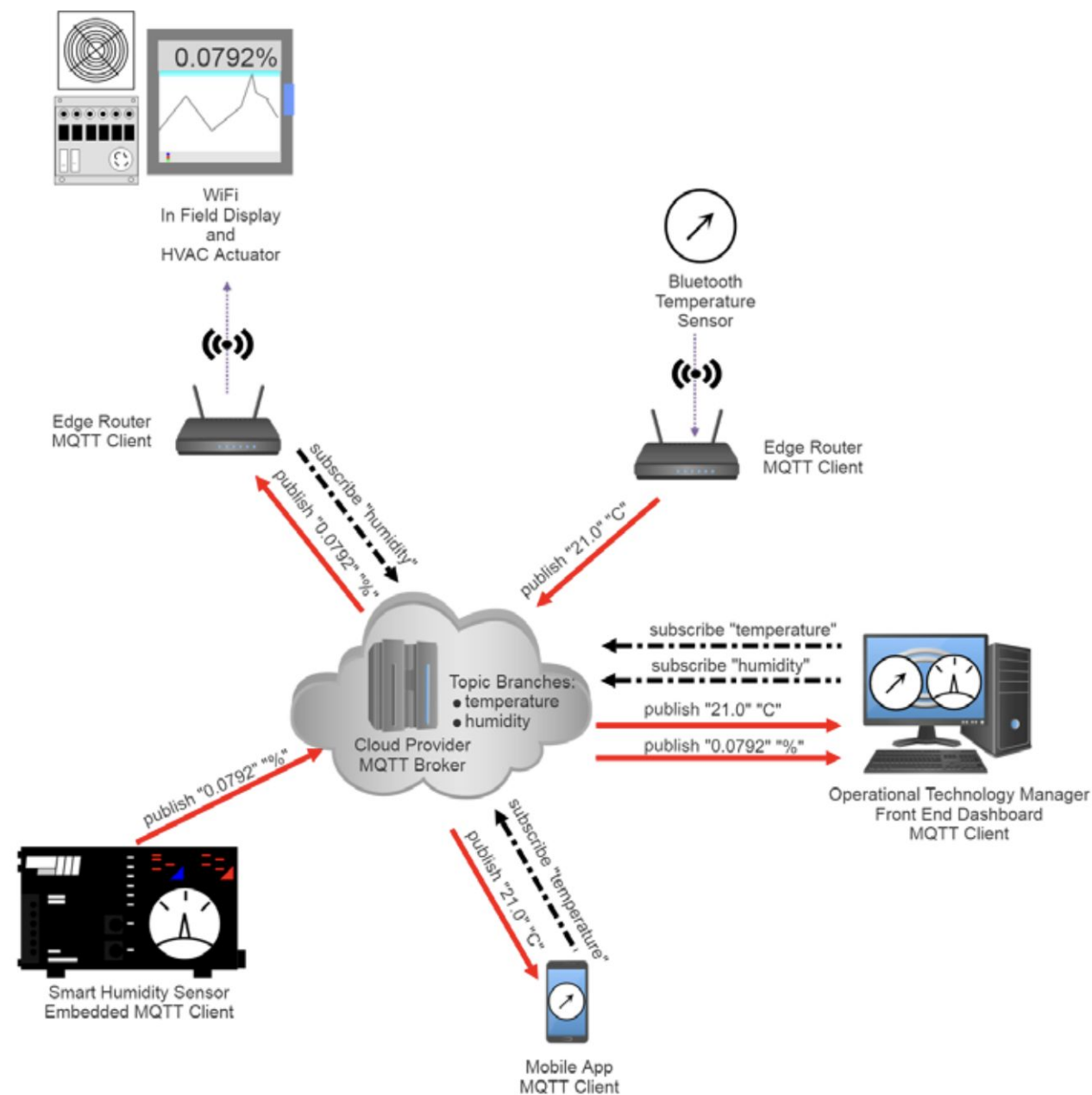


IA

MQTT [2]

Arquitetura

- clientes executam na borda
- tópicos: temperatura e umidade
- um cliente pode ser subscriber de vários tópicos





IA

MQTT: simples de implementar [1]

Connect

Subscribe

Publish

Unsubscribe

Disconnect

```
client = new Messaging.Client(hostname, port, clientId)
client.onMessageArrived = messageArrived;
client.onConnectionLost = connectionLost;
client.connect({ onSuccess: connectionSuccess });

function connectionSuccess() {
    client.subscribe("planets/earth");
    var msg = new Messaging.Message("Hello world!");
    msg.destinationName = "planets/earth";
    client.publish(msg);
}

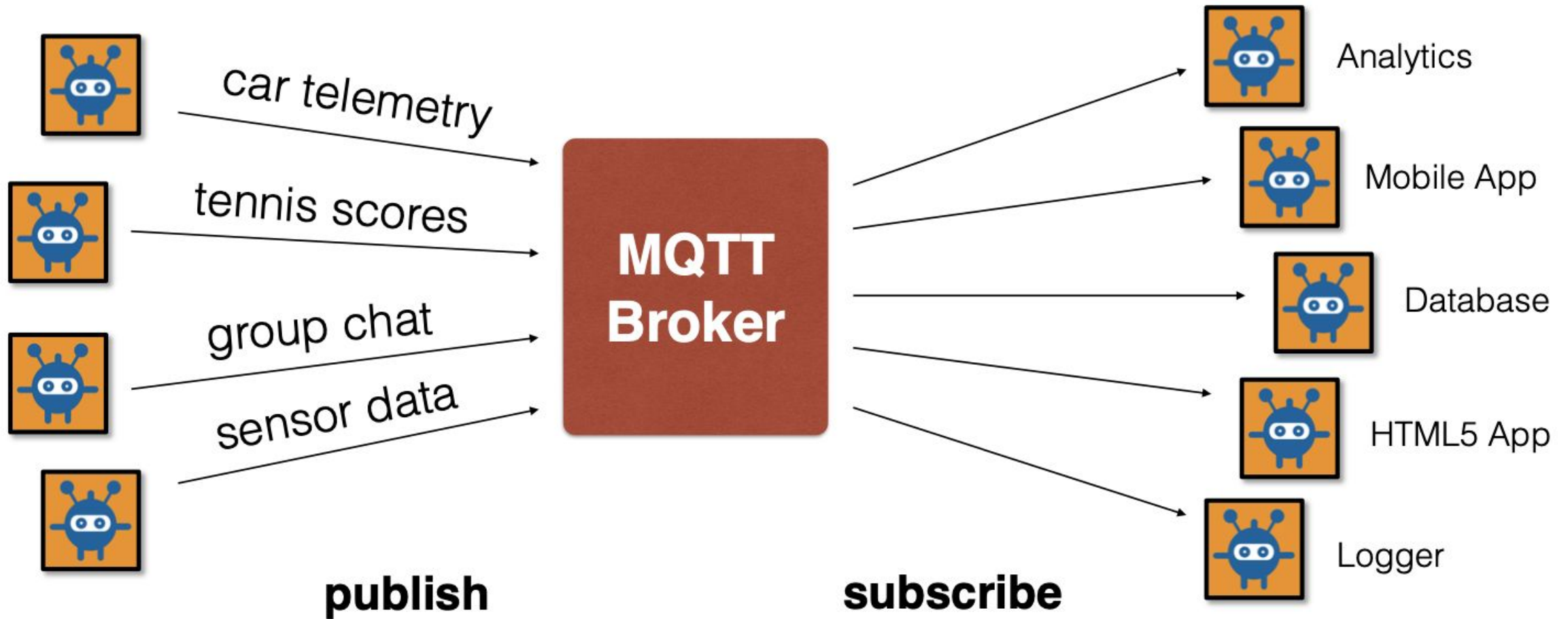
function messageArrived(msg) {
    console.log(msg.payloadString);
    client.unsubscribe("planets/earth");
    client.disconnect();
}
```

Eclipse Paho JavaScript MQTT client



IA

MQTT: pub/sub desacopla emissores de receptores [1]

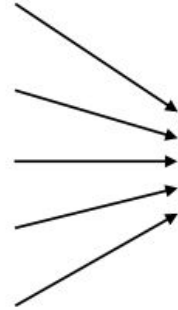




IA

MQTT: permite subscrições wildcard [1]

scores/football/big12/Texas
scores/football/big12/TexasTech
scores/football/big12/Oklahoma
scores/football/big12/IowaState
scores/football/big12/TCU
scores/football/big12/OkState
scores/football/big12/Kansas
scores/football/SEC/TexasA&M
scores/football/SEC/LSU
scores/football/SEC/Alabama



**MQTT
Broker**

scores/football/big12/Texas



Texas Fan

scores/football/big12/+



Big 12 Fan

scores/#



ESPN

single level wildcard: +

multi-level wildcard: #



IA

MQTT: projetado para tráfego de rede mínimo e dispositivos limitados [1]

- cabeçalho pequeno
 - PUBLISH: 2-4 bytes
 - CONNECT: 14 bytes
 - HTTP: 0,1-1KB
- payload binário (sem texto)
- clientes pequenos
 - 30 KB em C, 100 KB em Java
- trocas de mensagens mínimas
 - MQTT tem keep alive configurável (2 byte PINGREQ/PINGRES)
- eficiente para a vida da bateria
 - gasta menos energia que o HTTP para manter a conexão aberta, como também para receber e enviar mensagens



- cabeçalho padrão de 2 bytes
- cabeçalho variável (opcional)
- payload variável (opcional)

MQTT Packet Structure



MQTT Control Header



MQTT Control Packet Type

Name	Value	Direction	Description
Reserved	0	Forbidden	Reserved
CONNECT	1	Client to Server	Client connect request
CONNACK	2	Server to Client	Connect ACK
PUBLISH	3	Client to Server or Server to Client	Publish a message
PUBACK	4	Client to Server or Server to Client	Publish ACK
PUBREC	5	Client to Server or Server to Client	Publish receive (assured delivery part 1)
PUBREL	6	Client to Server or Server to Client	Publish receive (assured delivery part 2)
PUBCOMP	7	Client to Server or Server to Client	Publish complete (assured delivery part 3)
SUBSCRIBE	8	Client to Server	Client subscribe request
SUBACK	9	Server to Client	Subscribe ACK
UNSUBSCRIBE	10	Client to Server	Unsubscribe request
UNSUBACK	11	Server to Client	Unsubscribe ACK
PINGREQ	12	Client to Server	PING request
PINGRESP	13	Server to Client	PING response
DISCONNECT	14	Client to Server	Client disconnecting
Reserved	15	Forbidden	Reserved

MQTT Control Packet Flags

Control Packet	Fixed Header Flags	Bit 3	Bit 2	Bit 1	Bit 0
CONNECT	Reserved	0	0	0	0
CONNACK	Reserved	0	0	0	0
PUBLISH	Used in MQTT 3.1.1	DUP	QoS	QoS	RETAIN
PUBACK	Reserved	0	0	0	0
PUBREC	Reserved	0	0	0	0
PUBREL	Reserved	0	0	1	0
PUBCOMP	Reserved	0	0	0	0
SUBSCRIBE	Reserved	0	0	1	0
SUBACK	Reserved	0	0	0	0
UNSUBSCRIBE	Reserved	0	0	1	0
UNSUBACK	Reserved	0	0	0	0
PINGREQ	Reserved	0	0	0	0
PINGRESP	Reserved	0	0	0	0
DISCONNECT	Reserved	0	0	0	0



IA

MQTT: Qualidade de Serviço para mensageria confiável [1]

QoS 0 at most once



PUBLISH

- doesn't survive failures
- never duplicated

QoS 1 at least once



PUBLISH

PUBACK

- survives connection loss
- can be duplicated

MQTT
Broker

PUBLISH

PUBREC

PUBREL

PUBCOMP



QoS 2 exactly once

- survives connection loss
- never duplicated

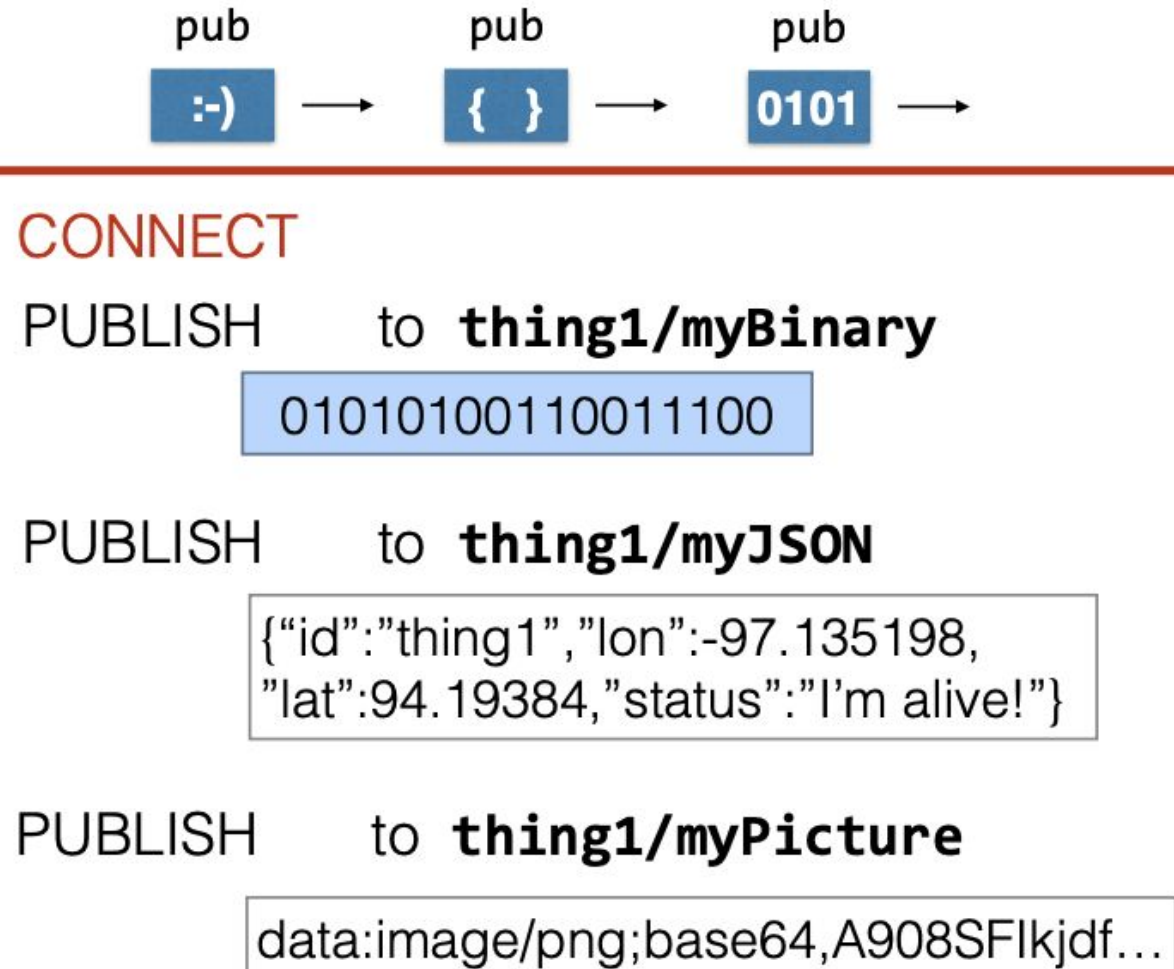


IA

MQTT: payload agnóstico para entrega flexível [1]



agnóstico:
qualquer tipo
de dados
(texto,
imagem,
audio, binário,
JSON, etc.)



**MQTT
Broker**



IA

MQTT: retém mensagens para os últimos valores em cache [1]



CONNECT ID=thing1 →
PUBLISH thing1/battery {"value":95} RETAIN →
PUBLISH thing1/battery {"value":94} RETAIN →
PUBLISH thing1/battery {"value":93} RETAIN →
DISCONNECT →



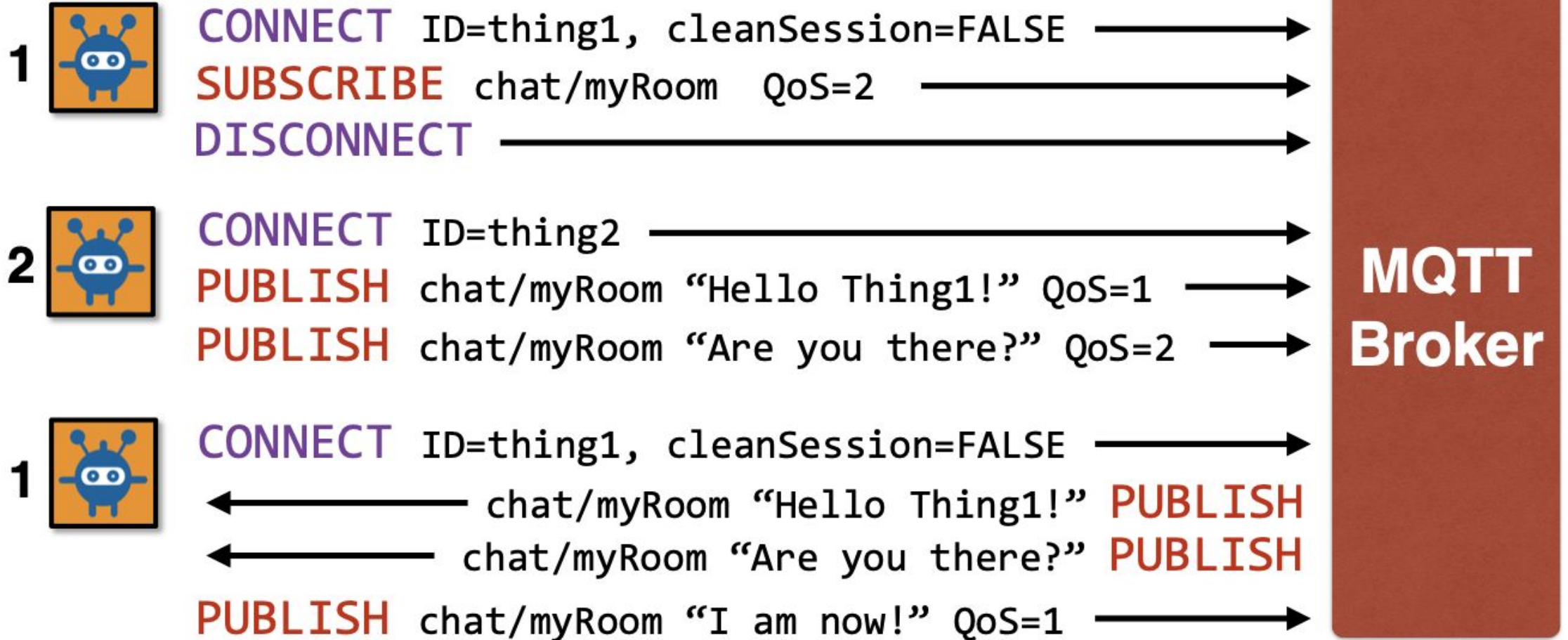
CONNECT ID=thing2 →
SUBSCRIBE thing1/battery →
← **RETAIN** thing1/battery {"value":93} **PUBLISH**

**MQTT
Broker**



IA

MQTT: client id e cleanSession para o estado da sessão [1]



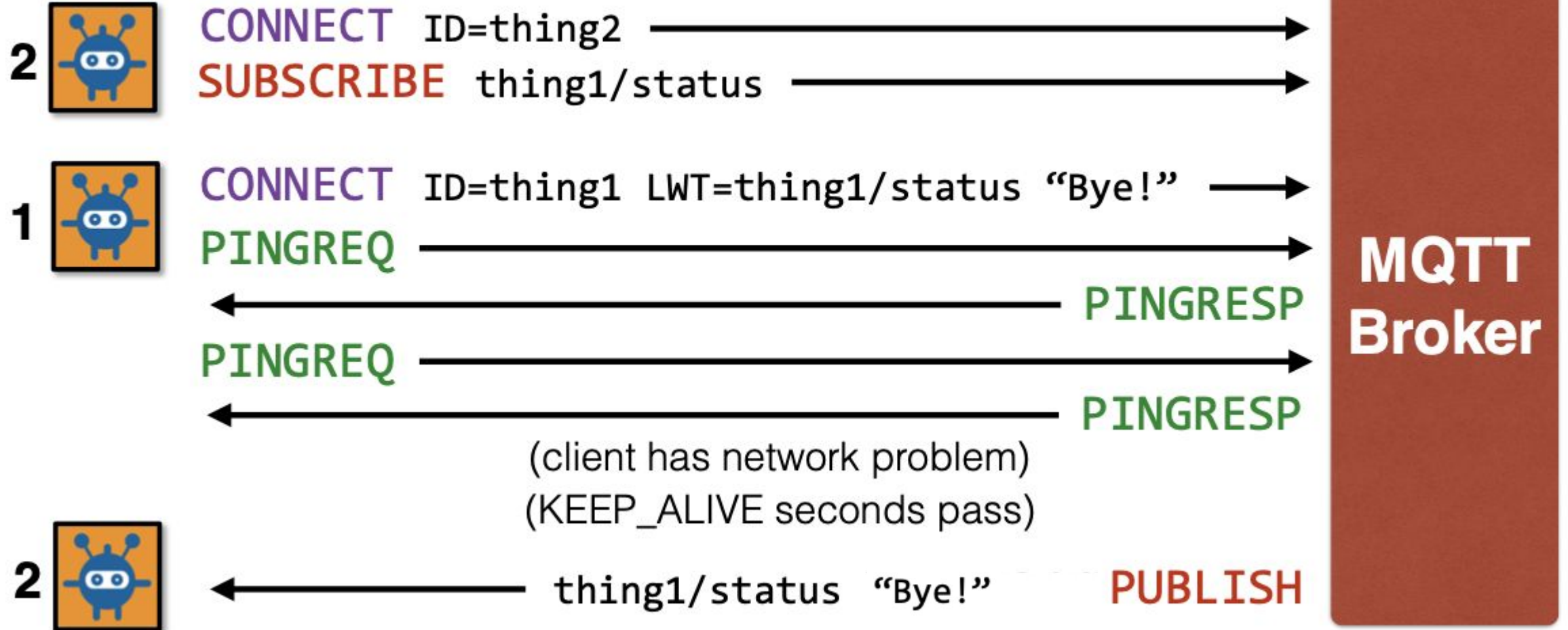


- clientID: identifica o cliente no servidor
- cleanSession:
 - 0: cliente e servidor devem salvar o estado da sessão antes do encerramento da conexão. Servidor faz um resumo para os subscribers.
 - 1: cliente e servidor começam uma nova conexão; descartam a sessão anterior



IA

MQTT: last will and testament [1]





- LWT é uma mensagem que um cliente especifica durante a fase de conexão
- Contém o tópico Last Will, QoS e a mensagem.
- Se o cliente se desconecta de um broker (ex. keep-alive timeout, erro /O, encerramento da conexão pelo cliente), o broker é obrigado a difundir a LWT para todos os subscribers daquele tópico
- Sistema keep-alive do MQTT
 - para clientes e broker terem certeza de que ainda podem usar a conexão
 - cliente envia mensagem PINGREQ que exige reconhecimento de uma mensagem PINGRESP.
 - timer presente no lado do cliente e broker
 - se uma mensagem não for transmitida dentro de intervalo de tempo predeterminado, o pacote keep-alive é enviado
 - se não houver recebimento quando o timer expirar, o broker envia o pacote LWT a todos os clientes



IA

MQTT: Segurança [1]



SSL/TLS

TCP/IP

CONNECT

with **username / password**

**MQTT
Broker**

- MQTT não define um modelo de segurança para autorização de conexão
- Brokers podem implementar o suporte para o TLS e políticas de conexão e mensageria
 - organiza tópicos por grupo e associar um username ao grupo
 - ex. bboyd está no grupo "IBM" e pode pub/sub IBM/bboyd/#

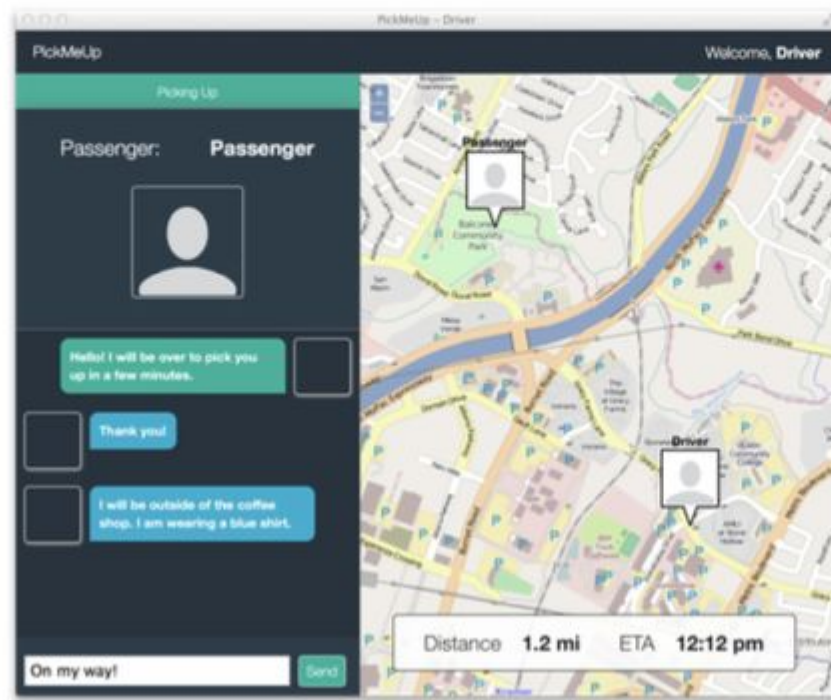


IA

MQTT: Estudo de Caso [1]

DEMO

PickMeUp!



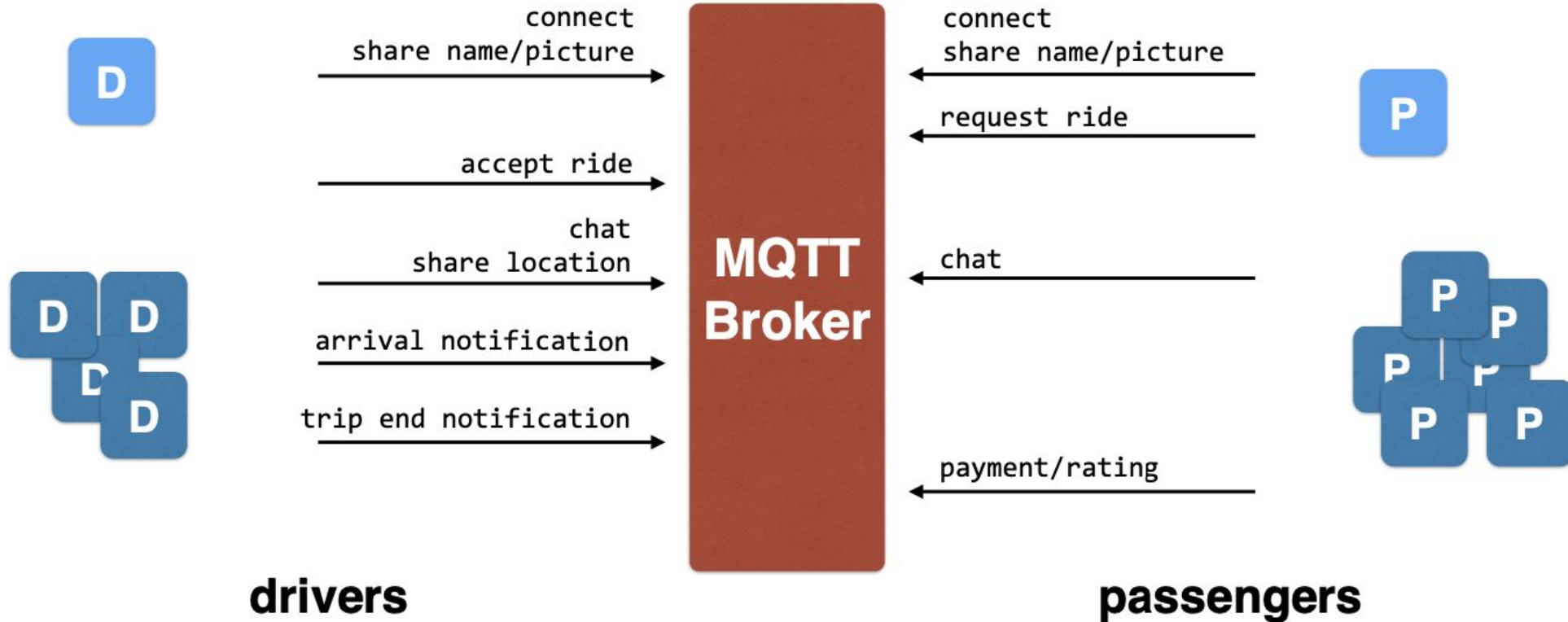


IA

MQTT: Estudo de Caso [1]

PickMeUp

Flow



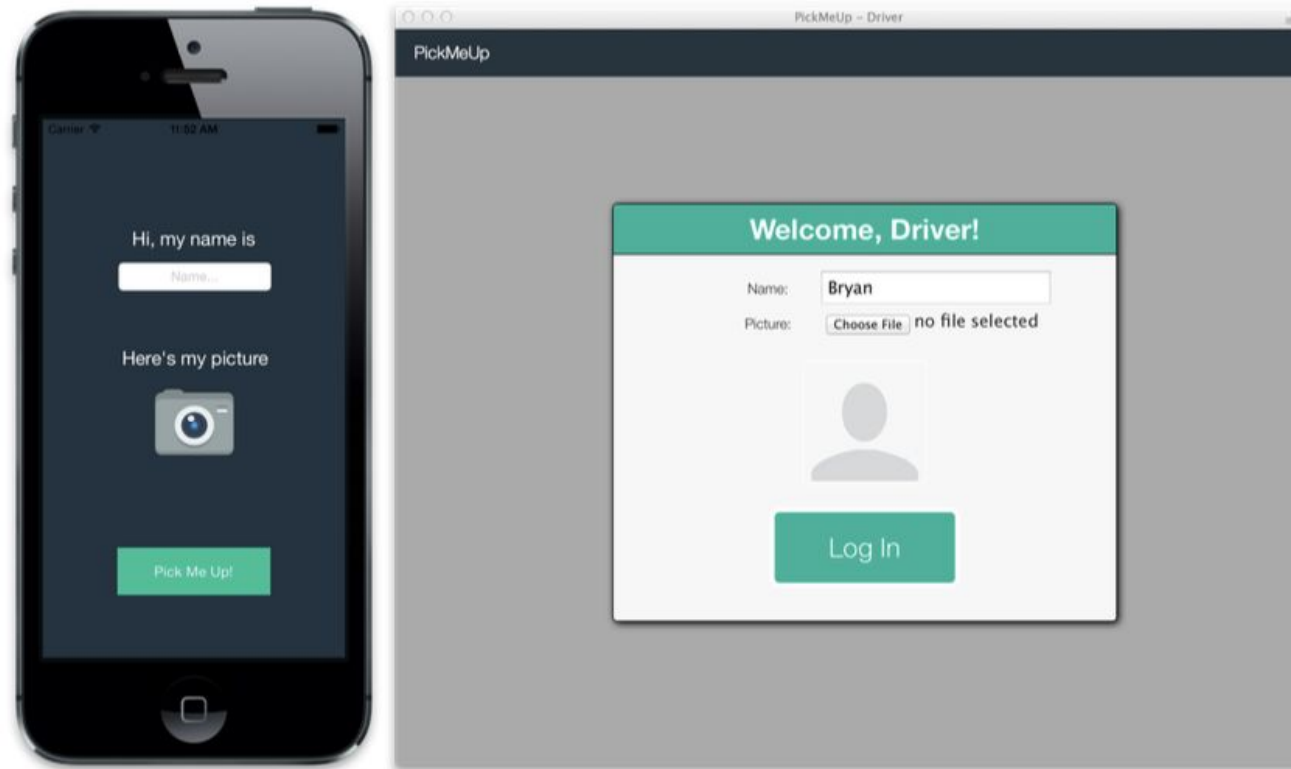


IA

MQTT: Estudo de Caso [1]

PickMeUp

Phase 1 — Connection





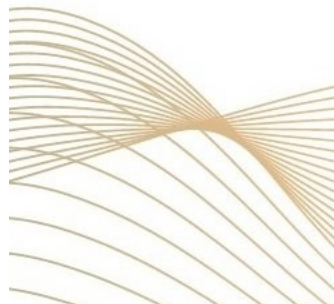
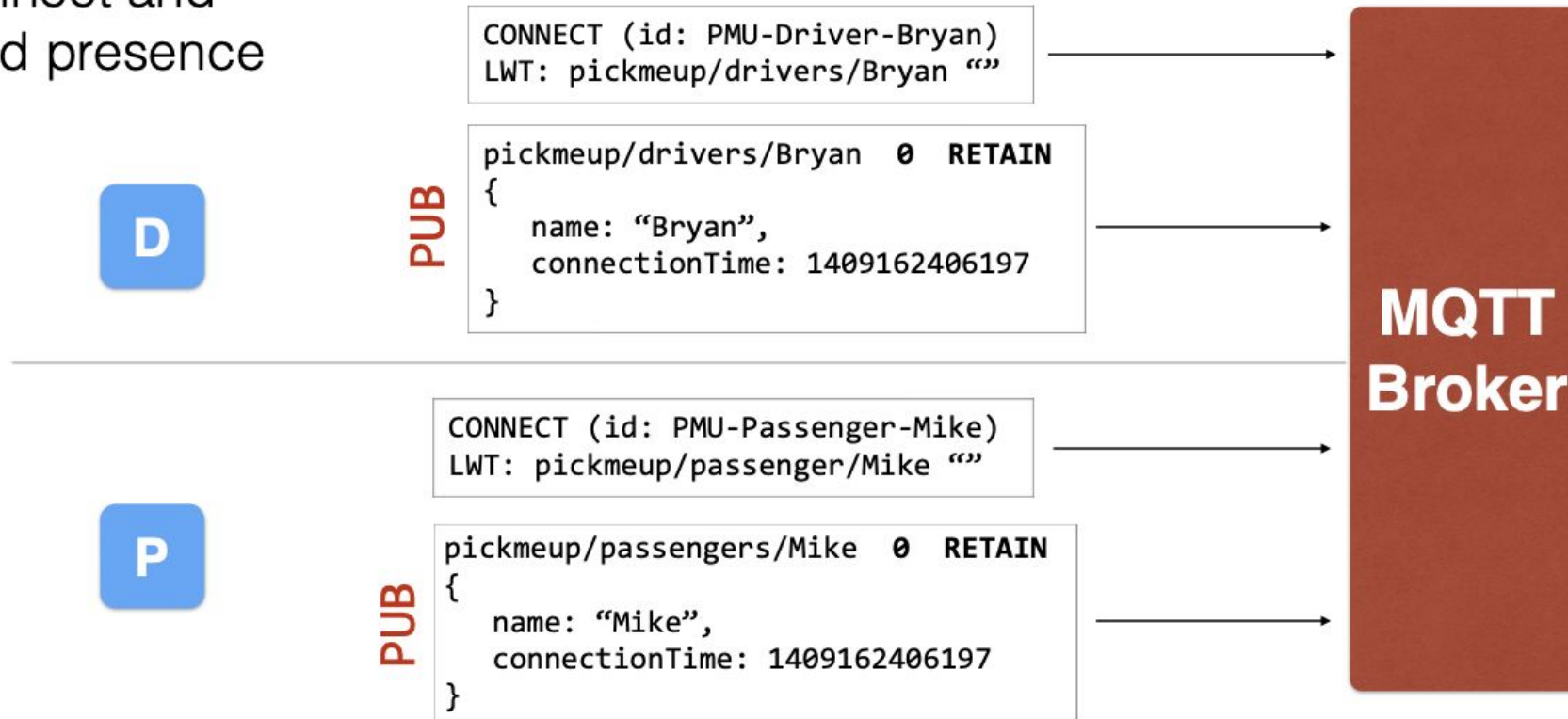
IA

MQTT: Estudo de Caso [1]

PickMeUp

Phase 1 — Connection

Connect and
send presence





IA

MQTT: Estudo de Caso [1]

PickMeUp

Phase 1 — Connection

Send picture,
subscribe to
inbox



PUB

```
pickmeup/drivers/Bryan/picture 0 RETAIN
{
  url: "data:image/png;base64,A198cf9013..."
}
```

SUB

```
pickmeup/drivers/Bryan/inbox 2
```

Send picture,
subscribe to
inbox



PUB

```
pickmeup/passengers/Mike/picture 0 RETAIN
{
  url: "data:image/png;base64,F87r19ZKa90..."
}
```

SUB

```
pickmeup/passengers/Mike/inbox 2
```

MQTT
Broker

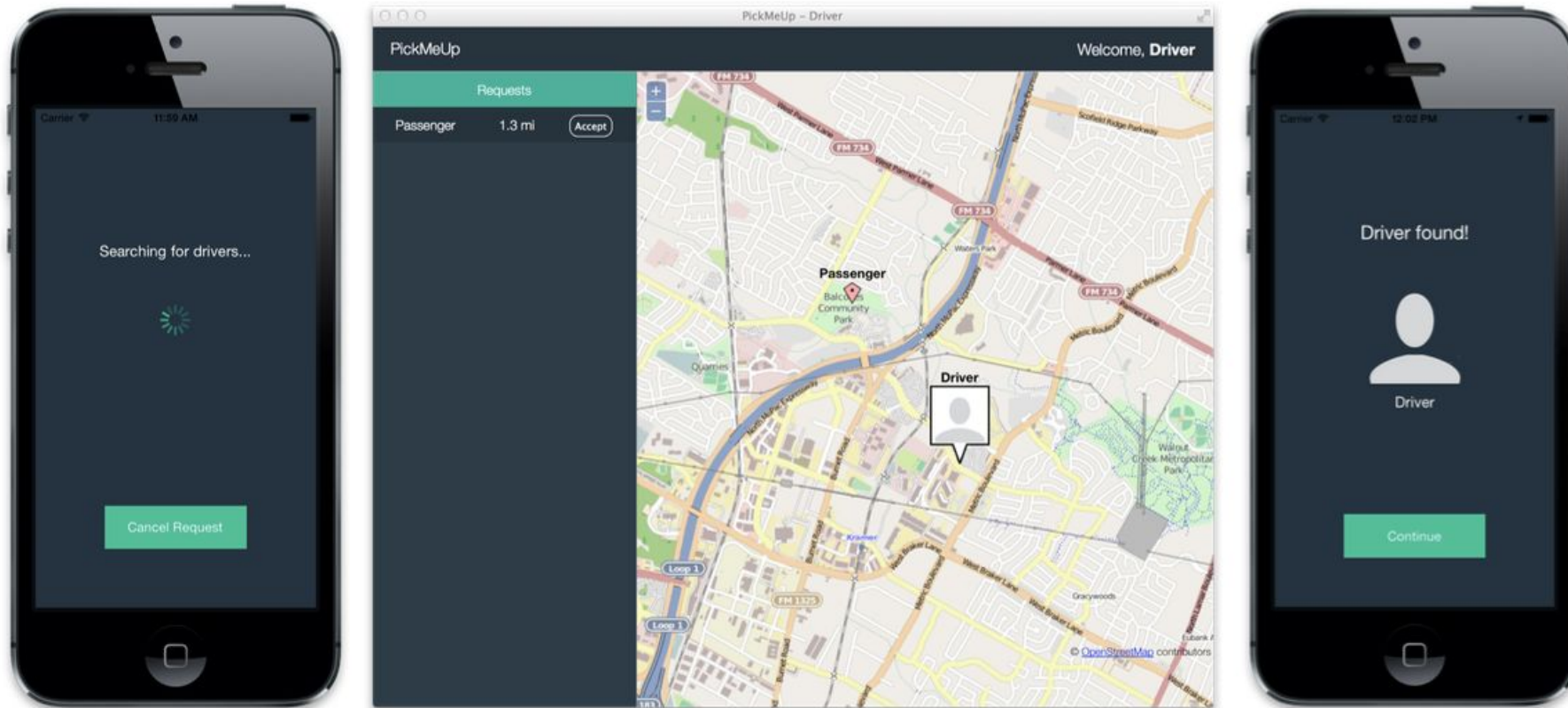


IA

MQTT: Estudo de Caso [1]

PickMeUp

Phase 2 — Pairing





IA

MQTT: Estudo de Caso [1]

PickMeUp

Phase 2 — Pairing

Subscribe to requests, accept request



SUB

```
pickmeup/requests/+ 0
```

PUB

```
pickmeup/passengers/Mike/inbox 1
{
  type: "accept",
  driverId: "Bryan",
  lon: <lon>, lat: <lat>
}
```

```
pickmeup/requests/Mike 0 RETAIN ""
```

Send request, subscribe to driver



PUB

```
pickmeup/requests/Mike 1 RETAIN
{
  name: "Mike", lon: <lon>, lat: <lat>
}
```

SUB

```
pickmeup/drivers/Bryan 0
```

```
pickmeup/drivers/Bryan/picture 0
```

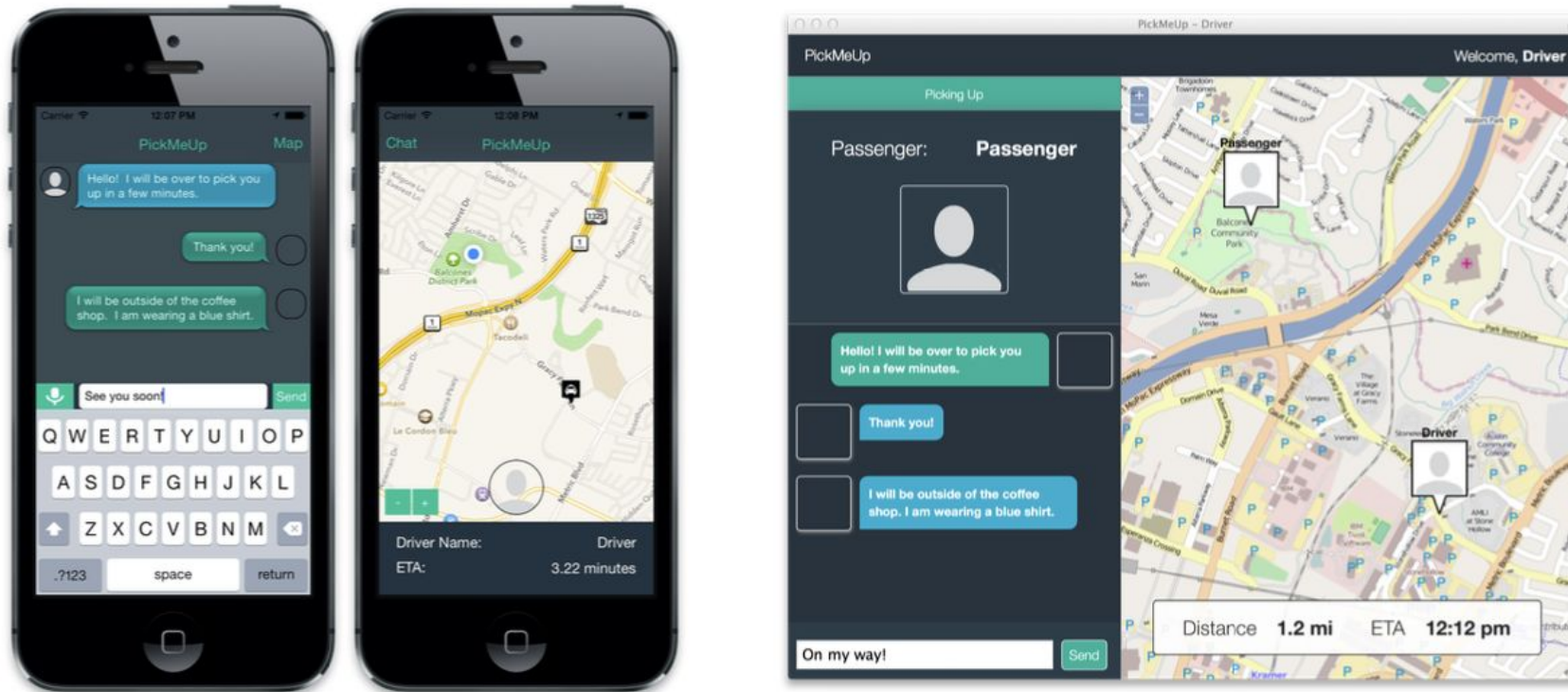
MQTT Broker



IA

MQTT: Estudo de Caso [1]

PickMeUp Phase 3 — Approaching





IA

MQTT: Estudo de Caso [1]

PickMeUp

Phase 3 — Approaching

Subscribe to
passenger data
chat to driver

Publish
driver location
chat to passenger



Driver

SUB

pickmeup/passengers/Mike 0

pickmeup/passengers/Mike/picture 0

pickmeup/passengers/Mike/location 0

pickmeup/drivers/Bryan/chat 0

PUB

```
pickmeup/passengers/Mike/chat 0
{
  format: "text", data: "On my way!"
  or
  format: "data:audio/wav;base64",
  data: "18bwagh0AH30913n..."
}
```

```
pickmeup/drivers/Bryan/location 0 RETAIN
{
  lon: <lon>, lat: <lat>
}
```

**MQTT
Broker**



IA

MQTT: Estudo de Caso [1]

PickMeUp Phase 3 — Approaching

Subscribe to
driver location
chat to passenger

SUB

```
pickmeup/drivers/Bryan/location 0
pickmeup/drivers/Bryan/chat 0
```

Publish
chat to driver

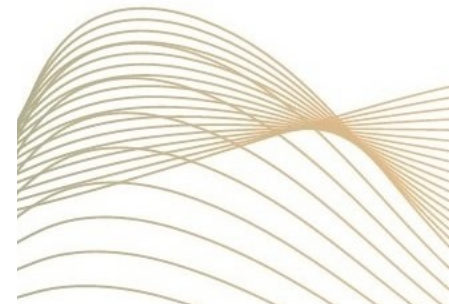
PUB

```
pickmeup/drivers/Bryan/chat 0
{
  format: "text", data: "On my way!"
  or
  format: "data:audio/wav;base64",
  data: "18bwagh0AH30913n..."
}
```

MQTT
Broker



Passenger



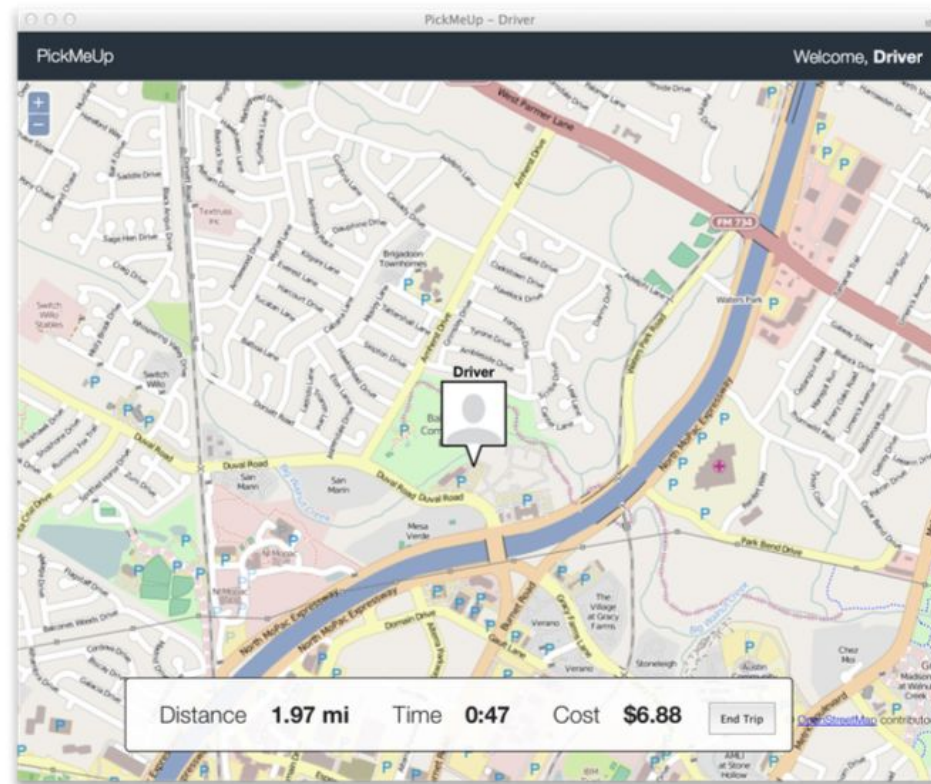


IA

MQTT: Estudo de Caso [1]

PickMeUp

Phase 4 — Driving





PickMeUp

Phase 4 — Driving

Publish
trip start notification
trip end notification



Driver

PUB

```
pickmeup/passengers/Mike/inbox 2
{
  type: "tripStart"
}
```

```
pickmeup/passengers/Mike/inbox 2
{
  type: "tripEnd",
  distance: 2.39,    // miles
  time: 178,         // minutes
  cost: 8.27         // dollars
}
```

**MQTT
Broker**

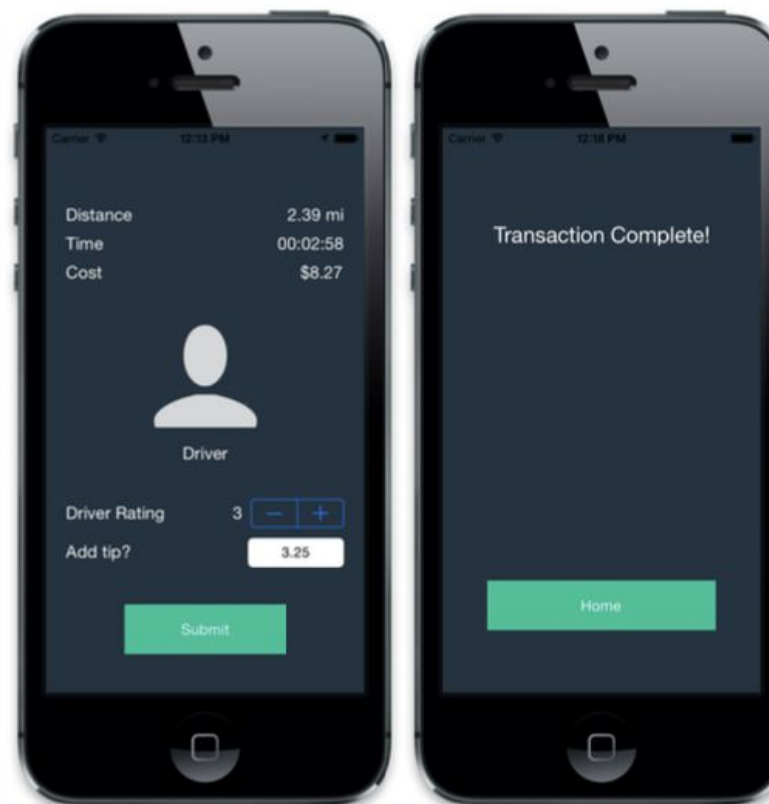
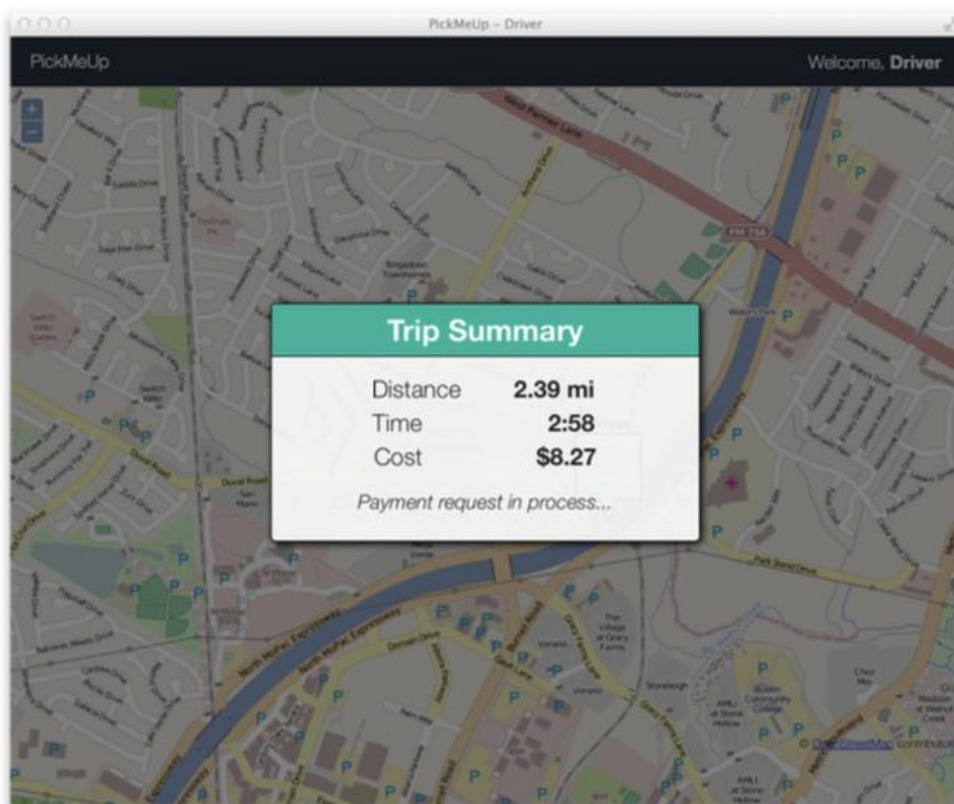


IA

MQTT: Estudo de Caso [1]

PickMeUp

Phase 5 — Payment





IA

MQTT: Estudo de Caso [1]

PickMeUp

Phase 5 — Payment

Publish rating and payment



PUB

```
pickmeup/payments 2
{
  driverId: "Bryan",
  passengerId: "Mike",
  cost: 8.27,
  rating: 3,
  tip: 3.25
}
```

Subscribe to payments, publish when processed



Backend

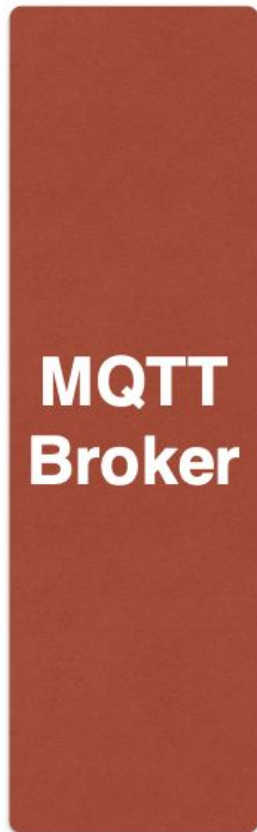
SUB

```
pickmeup/payments 2
```

PUB

```
pickmeup/passengers/Mike/inbox 2
{
  type: "tripProcessed",
  tip: 3.25, rating: 3
}
```

```
pickmeup/drivers/Bryan/inbox 2
{
  type: "tripProcessed",
  tip: 3.25, rating: 3
}
```





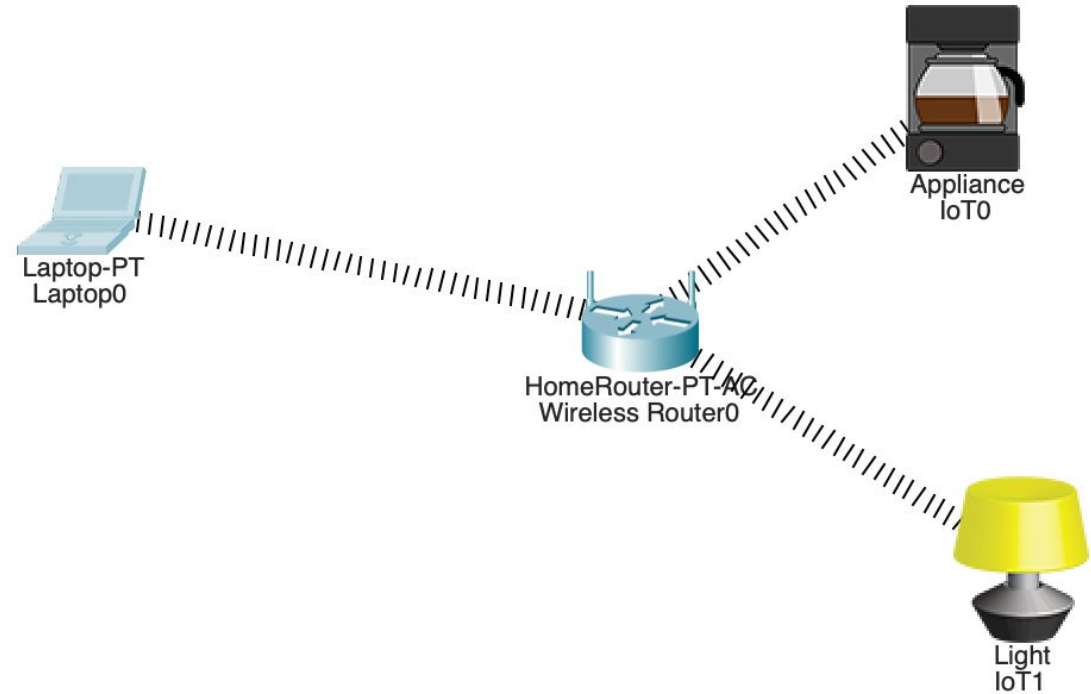
- publicar mensagens de presença retidas na conexão, usar o last will and testament para limpar
- usar mensagens retidas se você quer obter os dados dos últimos subscribers (posição do motorista, requisições)
- QoS 0 = atualização de informações, chat (coisas que podem ser perdidas)
- QoS 1 = requisições e aceites de requisições (importante, mas o cliente pode lidar com duplicações)
- QoS 2 = mensagens inbox, pagamento (importante, duplicações podem ser problemáticas)



IA

Demo-Lab: MQTT no Packet Tracer

- Broker: laptop
- pub/sub: coffee, lamp
- tópico:
 - coffee: {done, undone},
 - light: {on, off}





IA

Atv 11 - Dados de um sistema monitoramento de presença para armazenamento em nuvem

Nesta atividade, você deve listar todos os dados necessários para armazenamento em nuvem de um sistema de monitoramento de presença em uma Smart Home.

1 - Crie um documento de texto com uma tabela que relacione o nome do dado, tipo e descrição enfatizando a importância no armazenamento em nuvem.

2 - No final, envie o documento de texto para a Atv 11 do classroom.



IA

Referências Bibliográficas

- [1] B. Boyd. MQTT - A practical protocol for the Internet of the Things. 2014. Disponível em: <<https://www.slideshare.net/BryanBoyd/mqtt-austin-api>>. Acesso em: 20/04/2023.
- [2] Lea, P. IoT and Edge Computing for Architects - Implementing edge and IoT systems from sensors to clouds with communication systems, analytics, and security. 2ª ed. 2020.

Dúvidas?

Módulo de Internet das Coisas



UNIVERSIDADE
ESTADUAL DO CEARÁ



Instituto Iracema
PESQUISA E INOVAÇÃO



MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO

